

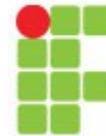


# LABORATÓRIO DE SISTEMAS OPERACIONAIS

PROF<sup>a</sup>. M.Sc. JULIANA HOFFMANN QUINONEZ BENACCHIO



- O Linux se refere aos itens de hardware como dispositivos (*devices*), e os referencia como arquivos.
- Os arquivos de dispositivo ficam no diretório /dev. Dentro desse diretório existem vários arquivos, um para cada dispositivo, além de subdiretórios que agrupam dispositivos de mesmo tipo.



- Exemplos de arquivos de dispositivo:
  - `/dev/fd0`: primeira unidade de disquete
  - `/dev/hda1`: primeira partição do primeiro disco rígido IDE
  - `/dev/sda1`: primeira partição do primeiro disco rígido SATA ou SCSI
  - `/dev/psaux`: mouse PS/2
  - `/dev/dsp`: placa de som



- `/dev/ttyS0`: primeira porta serial de comunicações, geralmente aquela utilizada pelo mouse serial (COM1)
- `/dev/usb/lp0`: primeira impressora ligada USB
- `/dev/video0`: primeira placa de captura de vídeo
- `/dev/lp0`: primeira porta paralela, geralmente usada por impressoras
- `/dev/null`: dispositivo nulo, usado para descartar saídas de programas;



- As distribuições GNU/Linux atuais possuem sistemas que criam automaticamente os arquivos de dispositivos para os novos itens de hardware que são instalados.
- Entretanto, isso só é válido para o hardware que possui suporte nativo no *Kernel* Linux. Os demais requerem uma instalação manual.



- O ***hotplug*** tem a função de detectar novos dispositivos.
- É responsável por carregar os módulos apropriados, ajustar permissões e executar tarefas diversas sempre que um novo dispositivo é conectado.



- Por exemplo, sem o *hotplug*, um pendrive pode funcionar normalmente, mas é necessário ter que carregar os módulos e montar o dispositivo sempre que quiser acessá-lo.
- Ou ainda, a impressora passa a aparecer no gerenciador de impressão apenas depois que você carrega os módulos apropriados e a configura manualmente e assim por diante.



- O *hotplug* surgiu originalmente como um *script* responsável por detectar e ativar dispositivos USB conforme eram plugados no micro.
- Embora tenha um funcionamento interno relativamente simples, o *hotplug* se mostrou bastante eficiente e passou a ser expandido para assumir a configuração de outros dispositivos.



- Atualmente, o *hotplug* não se limita a apenas pendrives, impressoras e câmeras, mas dá suporte a todo tipo de dispositivo *firewire*, PCMCIA, PCI e SCSI.
- A partir do *Kernel 2.6*, o *hotplug* se tornou um componente padrão do sistema.



- O diretório /proc é um sistema de arquivos virtual, pois ele não armazena arquivos reais no disco e sim informações do hardware, de configurações do sistema, de processos em execução na memória, dispositivos instalados no computador.
- Tudo criado na memória pelo *Kernel*. Por esse motivo ele também é conhecido como "*pseudo filesystem*"



- `/proc/version`: versão do sistema

```
#cat /proc/version
```



- `/proc/devices`: Lista de controladores de dispositivos (*device drivers*) configurados no *Kernel* em execução.

```
#cat /proc/devices | less
```



- `/proc/cpuinfo`: Informações sobre o processador instalado no sistema, como tipo, modelo e performance

```
#cat /proc/cpuinfo | less
```



- `/proc/meminfo`: Informações sobre a memória no sistema.

```
#cat /proc/meminfo | less
```



- `/proc/partitions`: Exibe as partições do sistema

```
#cat /proc/partitions
```



- **arch**: deriva de *architecture*, responsável por exibir qual a arquitetura do sistema. 32 ou 64 bits

**#arch**



- `lspci`: deriva de *list pci bus*, lista dispositivos conectados ao computador via barramento PCI. Obtém onde (qual barramento PCI) o hardware está localizado, o tipo de hardware, sua marca e modelo.

```
#lspci -v | less
```



- Opções [lspci]:
  - b**: exibe os IRQ's e os endereços conforme são vistos pelas placas no barramento PCI.
  - M**: aciona o mapeamento do barramento PCI, para localizar todos os dispositivos.
  - t**: exibe os barramentos PCI em formato de árvore
  - v**: exibe informações detalhadas sobre os dispositivos



- **lsusb**: deriva de *list usb port*, lista dispositivos USB conectados ao computador, exibindo seus respectivos fabricantes

```
#lsusb
```

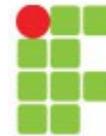


- Opções [lsusb]:
  - t: exibe os barramentos USB em formato de árvore
  - v: exibe informações detalhadas sobre os dispositivos



- **lshw**: deriva de *hardware lister*, fornece informações referentes ao conjunto de hardware que compõe a máquina.

```
#lshw | less
```



- Opções [lshw]:
  - html**: gera a saída em **html**
  - xml**: gera a saída em **xml**
  - short**: exibe um pequeno sumário
  - businfo**: gera saída com informações de barramento
  - X**: usa a interface gráfica (para esta opção é necessário o pacote **lshw-gtk**)



- Ações [lshw]:
  - c, -C, -class [classe]: exibe informações da classe especificada
  - disable [test]: não executa testes
  - enable [test]: executa testes
  - quiet: esconde a barra de status
  - sanitize: esconde informações confidenciais
  - numeric: exibe ids numéricos



- Classe [lshw]:

**address** – Endereços de memória para extensões de memória de vídeo ou ROM

**bridge** – PCI-to-PCI, AGP, PCMCIA

**bus** – Barramentos, sem nenhum hardware conectado

**communication** – Modem ou porta serial

**disk** – Drivers (incluindo físicos)

**display** – Componentes gráficos (sem o display)

**generic** – Outros componentes

**input** – Teclados, Mouses e Joystics

# Gerência de Dispositivos no Linux



INSTITUTO FEDERAL  
PARANÁ

**memory** – RAM, BIOS e *firmware*

**multimedia** – Som, TV e placas de vídeo

**network** – Bluetooth, Ethernet, FDDI e WLAN

**power** – Baterias e fontes de energia

**printer** – Impressoras e dispositivos multifuncionais

**processor** – Controladora de CPU e RAID

**storage** – Controladora IDE e SCSI

**system** – Tipo de sistema: laptop, Desktop, Servidor ou Computador

**tape** – Dispositivos DAT/DDS

**volume** – Sistemas de arquivos e partições



- `lsdev`: lista o hardware instalado no computador, especificando os endereços de E/S (Entrada/Saída), IRQ e canais DMA que cada dispositivo está utilizando.

```
#lsdev
```

```
#apt-get install procinfo
```